

-continued

```

<LICENSE HREF="http://www.fsc.com/coolestapp/license.html"/>
<!--FSC's CoolestApp is implemented in native code -->
<IMPLEMENTATION>
  <OS VALUE="WinNT"><OSVERSION VALUE="4,0,0,0"/></OS>
  <OS VALUE="Win95"/>
  <PROCESSOR VALUE="x86"/>
  <LANGUAGE VALUE="en"/>
  <CODEBASE HREF="http://www.fsc.org/coolestapp.cab"/>
  <!--CoolestApp needs CoolerApp -->
  <DEPENDENCY>
    <CODEBASE HREF="http://www.fsc.org/coolapp.osd"/>
  </DEPENDENCY>
</IMPLEMENTATION>
</SOFTPKG>

```

Had the CoolApp manifest file been stored in a cabinet distribution unit file along with the CoolApp components, the location of the distribution unit file would have been <http://www.fsc.org/coolapp.cab>.

In a second exemplary embodiment of the invention for purposes of this section, components contained in a distribution unit file are caused to be installed by OSD tags embedded on a Web page. If Fred's Software Company's Web page requires additional software to be downloaded and installed for viewing the page, FSC can use the OSD vocabulary within HTML commands to have the user's browser download the necessary components as shown in the two examples below.

```

<OBJECT CLASSID="clsid:9DBAFCF-592F-101B-85CE-00608CEC297B"
  VERSION="1,0,0,0"
  CODEBASE="http://www.fsc.com/coolestapp.osd"
  HEIGHT=100 WIDTH=200>
</OBJECT>
-or-
<APPLET code=myapplet.class id=coolestapp width=320 height=240>
  <PARAM NAME=useslibrary VALUE="coolestapp">
  <PARAM NAME=useslibraryversion VALUE="1,0,0,0">
  <PARAM NAME=useslibrarycodebase VALUE=
    "http://www.fsc.com/coolestapp.osd"
">
</APPLET>

```

The HTML <OBJECT> or <APPLET> tag informs an OSD-aware client browser, such as Microsoft Explorer 4, that there is additional software required to view the Web page. The browser invokes the package manager to execute the software package if it is already installed or to install it if not. If not already installed, the package manager instructs the browser to download the distribution file unit and proceeds with the installation as described in the previous section. The "CODEBASE" element in <OBJECT> and the "useslibrarycodebase" tag in <APPLET> can point to the manifest file or to the distribution unit file.

In a third exemplary embodiment of the invention for purposes of this section, a distribution unit file is used to automatically distribute software from Fred's Software Company's server to the user's computer. This automatic distribution across a network employs "channels" to which the user subscribes to automatically "push" software components through a client agent such as a browser. The channel is described using a Channel Definition Format (CDF) which is also based on XML. A CDF file uses the OSD elements to inform a CDF-aware client agent as to what software components should be downloaded and installed.

```

<CHANNEL HREF="http://www.fsc.com/intropage.htm">
<SELF="http://www.fsc.com/software.cdf"/>
<TITLE>A Software Distribution Channel</TITLE>
<SOFTPKG
  HREF="http://www.fsc.com/aboutsoftware.htm"
  AUTOINSTALL="yes"
  NAME="{D27CDB6E-AE6D-11CF-96B8-444553540000}"
  VERSION="1,0,0,0">
  <IMPLEMENTATION>
    <OS VALUE="WinNT"><OSVERSION VALUE="4,0,0,0"/></OS>
    <OS VALUE="Win95"/>
    <PROCESSOR VALUE="x86"/>
    <CODEBASE HREF="http://www.fsc.com/coolestapp.cab"/>
  </IMPLEMENTATION>
</SOFTPKG>
</CHANNEL>

```

This section has described a particular implementation of the package manager which is directed to install software by OSD elements embedded in an XML document. The processing of a manifest file described in previous section when written as XML document is described. In addition, alternate embodiments in which a separate XML document resides on a Web page to direct a browser to invoke the package manager to install a software package is also described in this section.

Conclusion

A software package manager has been described which manages the installation, execution and uninstallation of software packages acquired through various media. The software manager uses a manifest file, a distribution unit, and a code store data structure to accomplish its functions. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention.

For example, those of ordinary skill within the art will appreciate that the file and data structures described herein can be easily adapted to future distribution media. Furthermore, those of ordinary skill within the art will appreciate that future extensible languages which are platform and operating system independent can be used to direct the software package managers actions.

The terminology used in this application with respect to is meant to include all hardware and software platforms. Therefore, it is manifestly intended that this invention be limited only by the following claims and equivalents thereof.

We claim:

1. A computerized method for installing a software package on a computer over a network with reference to a code store data structure tracking whether components of software packages are installed on the computer, comprising:
 - a) downloading to the computer from the network a manifest file that describes at least one distribution unit for a software package, wherein the manifest file specifies a non-local location for acquiring a distribution unit when not present on the computer;
 - b) resolving software dependencies on the computer for the software package by consulting the code store data structure and the manifest file to identify at least one distribution unit indicated in the manifest file as depended on by the software package but not yet installed on the computer;